Computer Science Department

TECHNICAL REPORT

"Queueing Delays in Buffered Multistage Interconnection Networks"

by

C. Bouras[†]
J. Garofalkis[†]
P. Spirakis[‡]
V. Triantafillou[†]

Technical Report #289
March, 1987

NYU COMPSCI TR-289 c.1
Bouras, C
Queueing delays in buffered
multistage interconnection
networks.

NEW YORK UNIVERSITY

Department of Computer Science
Courant Institute of Mathematical Sciences
251 MERCER STREET, NEW YORK, N.Y. 10012

"Queueing Delays in Buffered Multistage Interconnection Networks"

by

C. Bouras†
J. Garofalkis†
P. Spirakis‡
V. Triantafillou†

Technical Report #289
March, 1987

† Computer Technology Institute Greece and Computer Engineering Dept., Patras University, Greece.

‡ Courant Institute of Mathematical Sciences, NYU, USA.

This research was supported in part by the Ministry of Industry, Energy and Technology of Greece.

Abstract

Our work deals with the analysis of the queueing delays of buffered multistage Banyan networks of multiprocessors. We provide tight upper bounds on the mean delays of the second stage and beyond, in the case of infinite buffers. Our results are validated by simulations performed on a network simulator constructed by us. The analytic work for network stages beyond the first, provides a partial answer to open problems posed by previous research.

1. Introduction

With the realistic appearance of low cost microprocessors, the multiprocessor architectures became very attractive. The principal characteristics of such systems are the massive parallelism and the ability of each processor to share a single main memory or a set of memory modules. This sharing capacity is provided through an interconnection network between the processors and the memory modules. Among the different physical forms available for the network are the time-shared bus, the crossbar switch and the multistage packet switching networks, such as the Banyan and Delta networks (see e.g. [GL, 73] and [P, 81]). The bus has a very limited transfer rate, especially in the case of machines with thousands of processors. The full crossbar is not only very expensive but it also requires a tremendous amount of interconnections which clashes with engineering limitations. The multistage networks are intermediate cases, with small cost and good performance. They have been adopted recently by the industry (e.g. the RP3 machine of IBM, [G, 84]). Since the network is an important component of the multiprocessor machine, it is important to have a solid understanding of its performance. In most of the proposed designs the network supports dynamic access from each processor to each memory module, and the traffic through the network consists of short items (requests to memory and replies). The requests are being dynamically generated independently at each processor. The pattern of requests is essentially random and varies rapidly.

In this paper we analyze the queueing delays of buffered multistage interconnection networks. Previous analyses have been reported in [KS, 83] and [KSW, 84].
They mostly concentrate on the analysis of the delays in the first stage and provide the delay distribution under uniform and nonuniform traffic. The delays at subsequent stages are estimated there by simulations and there are several conje-

ctures about them. The main obstacle is that the distribution of packet arrivals is not time-independent from the second stage on, even if such independence is assumed for the first stage.

Our work provides analytical tight upper bounds for the delays at the second stage and beyond. It indicates that delays tend to increase from the first to the successive stages. It also indicates that (for any fixed packet generation rate) after the second stage there is no notable difference between the delay times, thus giving a partial positive answer to the conjecture and experimental results of [KS, 83]. Our results are backed up by the simulations we did on a self-driven network simulator.

2. Networks and modelling assumptions

The networks we consider are built of switches connected by unidirectional lines.(In actual parallel computers it is necessary to send replies back through the network. There are many ways to change definitions to allow that). A k input, k output switch can receive packets at each of its k input ports and send them through each of its k output ports. A network (see [KS, 83]) is a labelled digraph where nodes are of the following three types:

(i) source nodes (indegree 0, outdegree 1)
(ii) sink nodes (indegree 1, outdegree 0)
(iii) switches (positive indegree and outdegree).

Each edge represents one or more lines going from a node to its successor. A banyan network (see [GL, 73]) is a network with a unique path from each source to each sink node. An n-stage banyan network is a banyan network whose nodes can be arranged in stages, with all the source nodes connected to switches at the first stage, and all the outputs at stage i connected to inputs at stage i+1. An n-stage rectangular banyan network of degree k is an n-stage banyan network built of kxk switches.

We restrict our analysis to oblivious routing algorithms, i.e. algorithms in which the path of a packet through the network is fixed at the source node issuing it. The path can be encoded as the sequence of labels of the successive switch outputs of the path (path descriptor).

Our modelling assumptions are those usually used in the literature ([KS, 83],[P,81]). Packets are generated at each source node by independent, identically distributed random processes. Each processor generates with probability p at each cycle a packet, and sends a generated packet with equal probability to any sink node (uniform access). The network is assumed to be synchronous (discrete-time) so that packets can be sent only at times t_c, $2t_c$, $3t_c$,... where t_c is the network cycle time. We assume that t_c is also the cycle time of each switch. For the analysis, without loss of generality, we will take $t_c=1$. The above assumptions imply the following lemma.

Lemma 1 [KS, 83]

Let packets be generated at the source nodes of a banyan network by independent, identically distributed random processes, that uniformly distribute the packets over all of the sink nodes. Assume that the routing logic at each switch is fair, i.e. that conflicts are randomly resolved. Then

(a) The patterns of packet arrivals at the inputs of the same switch are independent.

(b) Packets arriving at an input of a switch are uniformly distributed over the outputs of that switch.

(c) For uniform networks, for each stage, the pattern of packet arrivals at the inputs of that stage have the same distribution.

The uniform access assumption allows us to represent any kxk switch as a system of k queues working in parallel, with a deterministic server each (of service time equal to 1). Any packet entering any of the k inputs of the switch, goes with probability $\frac{1}{k}$ to any of the (output) queues of the switch.

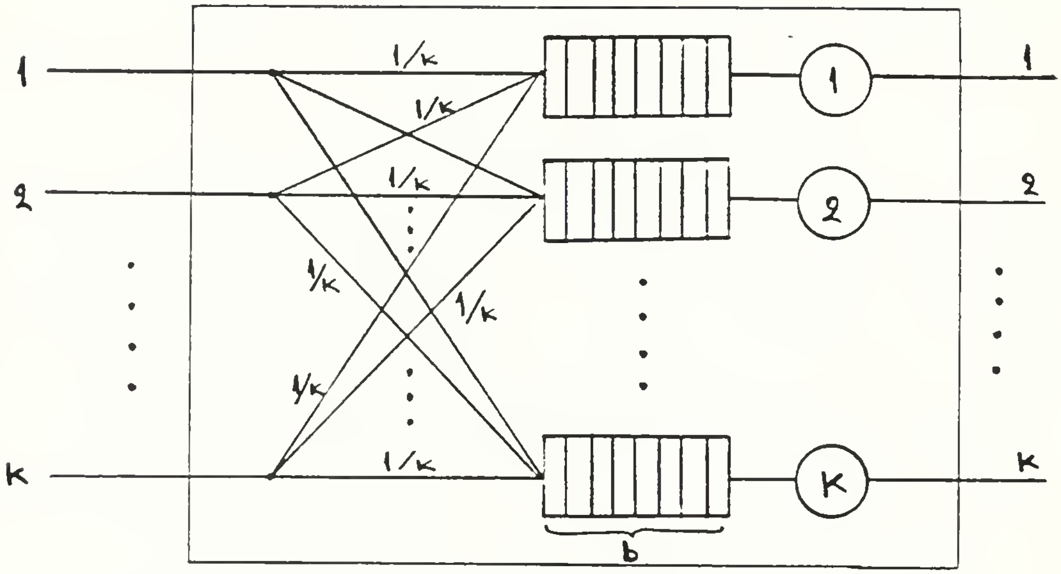

Figure 1: A kxk switch model with buffer size $b \geq 0$.

3. The isolated kxk switch and the B/D/1 queue.

Definition A B/D/1 queue is a queue whose input (arrival) process is a Bernoulli process B(q,N) where N is the number of trials and q is the success probability, i.e. where the Prob{x packets arrive at the queue at the next unit of time}=$\binom{N}{x}q^{x}(1-q)^{N-x}$. The server of the queue is deterministic (of service time equal to 1) and the arrivals are considered to occur just before the end of the corresponding time unit. (Discrete time queue). We consider the length of each time unit to be equal to 1.

It is clear that any given output queue of the isolated kxk switch can be modelled by a $B(\frac{p}{k},k)$ queue, since with probability p/k a packet both appears in a given input and moves to the queue under discussion. In fact, if we consider an isolated kxk switch (or any switch of the first stage of the network) and if v_i=number of arrivals at (output) queue i of the switch, for i=1,2,...,k, then the total number of arrivals in the inputs of the switch during a cycle, is

$$v=v_1+\ldots+v_k \qquad \text{and}$$
$$\text{Prob}\{v=n\}= \binom{k}{n} p^n (1-p)^{k-n} \quad \text{if } 0\le k\le n$$
$$= 0 \quad \text{else}$$

Since any assignment of n arriving concurrently messages to the output queues has probability $(\frac{1}{k})^n$ (by the assumption of random resolution of conflicts), we conclude that the joint arrival distribution at the k queues is

$$\text{Prob}\{v_1=a_1, v_2=a_2, \ldots, v_k=a_k / a_1+\ldots+a_k=n\}=$$

$$= \frac{n!}{a_1!a_2!\ldots a_k!} \left(\frac{1}{k}\right)^n \qquad \text{(EQ 1)}$$

It is easy to prove that

Lemma 2 (Due to M. Snir, proof in the full paper)

The marginal densities $\text{Prob}\{v_i=a_i\}$ of (EQ 1) are the same with the Bernoulli $B(\frac{p}{k},k)$, assumed in the analysis of the B/D/1 queue.

Definition Let q_n be the number of packets in the B/D/1 queue with input B(p/k,k) at the end of cycle n.

Definition Let v_n be the number of packets joining the B/D/1 queue of input $B(\frac{p}{k},k)$ at cycle n.

Definition Let $\Delta(k)=0$ if $k\le 0$ and $\Delta(k)=1$ else.

It is clear that

$$q_{n+1}=q_n-\Delta(q_n)+v_{n+1} \qquad \text{(EQ 2)}$$

Assuming a steady state distribution $\tilde{q}=\lim_{n\to\infty} q_n$ (which always exists if $p<1$) we have (by taking means)

$$E(\tilde{q})=E(\tilde{q})-E(\Delta(\tilde{q}))+E(\tilde{v})$$

implying

$$E(\Delta(\tilde{q}))=E(\tilde{v}) \qquad \text{(EQ 3)}$$

But $\tilde{v}$ has the same distribution with v_n (since the arrival process *is not time-dependent*) i.e.

$$E(\tilde{v})=E(v_n)=k\frac{P}{k}=P \qquad \text{(EQ 4)}$$

However

$$E(\Delta(\tilde{q}))=\sum_{k=0}^{\infty}\Delta(k)\mathrm{Prob}(\tilde{q}=k)=\mathrm{Prob}(\tilde{q}>0) \qquad \text{(EQ 5)}$$

Comparing EQ 3, EQ 4, EQ 5 we have

Lemma 3 The utilization of the B/D/1 queue of input $B(p/k,k)$ is equal to p. $\square$

We can also easily prove that

Lemma 4 The steady state mean queue length of the B/D/1 queue (i.e. of *any output queue of the first stage* of the network) is

$$E(\tilde{q})=p+\frac{p^2(1-1/k)}{2(1-p)} \qquad \text{(EQ 6)}$$

(The above equation includes the packet in service).

Proof sketch

EQ 6 was derived in [KS, 83]. The crucial steps to get it is (1) the method of moments of [KLEI, 75] which proceeds by squaring EQ 2 and then taking means and limits, and (2) the fact that v_{n+1} is independent of v_i, $i<n+1$, (and thus of q_n), only because of the assumption about packet generation in the processors. $\square$

It is an easy corollary of the above that the mean delay in the first stage is

$$W_{queue}=\frac{p(1-\frac{1}{k})}{2(1-p)}, \text{ in number of cycles.}$$

4. The queuing delays of the second stage.

By symmetry, it is enough to analyze the first output queue, Q_1, of a particular switch S of the second stage.

Let β_1^n be the number of customers in Q_1 at cycle n.

Let x_i^{n+1} be the (possible) arrival of a packet from input i to Q_1, during cycle n+1.

Let $q_{11}^n, \ldots, q_{1k}^n$ be the number of customers in the queues (of the first stage) which feed the switch S containing Q_1, during cycle n. (See Figure 2)

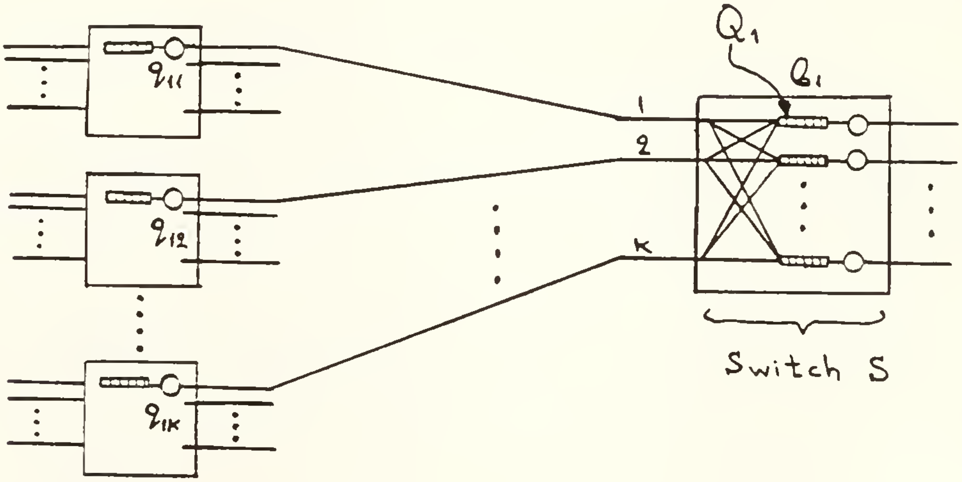

Figure 2: The second stage.

It is clear that

$$\beta_1^{n+1}=\beta_1^n-\Delta(\beta_1^n)+(x_1^{n+1}+\ldots+x_k^{n+1}) \qquad \text{(EQ 7)}$$

where for each i

$$x_i^{n+1}=\Delta(q_{1i}^n) \quad \text{with probability } \frac{1}{k}$$
$$= 0 \quad \text{else}$$

By Lemma 1, the $x_1^{n+1}, \ldots, x_k^{n+1}$ are idependent of each other, also the $q_{11}^n, \ldots, q_{1k}^n$ are independent of each other.

However each x_i^{n+1} depends on each of q_{1j}^n and thus the β_1^n and x_i^{n+1} are not independent random variables.

Definition

Let y_i^n= the size of β_1^n given that $x_i^{n+1}=1$.

Let $\varphi_i^n(a)=\text{Prob}\{y_i^n=a\}$

We need a rather technical lemma:

Lemma 5 For each i,n we have:

$$E(\beta_1^n \cdot x_i^{n+1}) = \mathrm{Prob}\{x_i^{n+1}=1\} \cdot E(y_i^n)$$

and

$$E(\Delta(\beta_1^n) \cdot x_i^{n+1}) = \mathrm{Prob}\{x_i^{n+1}=1\} \cdot (1-\varphi_i^n(0))$$

Proof sketch

$$E(\beta_1^n \; x_i^{n+1}) = \sum_{\alpha\;\beta} \alpha \cdot \beta \cdot \mathrm{Prob}\{\beta_1^n = \alpha \text{ and } x_i^{n+1} = \beta\}$$

$$= \sum_{\alpha \geq 0} \alpha \cdot \mathrm{Prob}\{\beta_1^n = \alpha \text{ and } x_i^{n+1} = 1\}$$

$$= \sum_{\alpha > 0} \alpha \cdot \mathrm{Prob}\{x_i^{n+1}=1\} \mathrm{Prob}\{\beta_1^n = \alpha / x_i^{n+1}=1\}$$

QED

Similarly for the second equation ☐

Let us also note that $x_i = \lim_{n\to\infty} x_i^n = \Delta_{\tilde{q}_{1i}}$ with prob 1/k

$= 0$ else,

hence $E(x_i) = \mathrm{Prob}\{x_i > 0\} = \frac{p}{k}$ (by Lemma 3) $\forall$ i and thus $E(x_1+\ldots+x_k) = p$, implying that $\lim_{n\to\infty} \beta_1^n$ exists if $p<1$.

By taking, then, limits and expectations in (EQ 7) we conclude

$$E(\Delta(\beta_1)) = E(x_1+\ldots+x_k) = p \qquad \text{i.e.}$$

Lemma 6

The utilization of the queue Q_1 of S is p, at the steady state. ☐

The above also imply that $y_i = \lim_{n\to\infty} y_i^n$ exists if $p<1$. By squaring (EQ 7) (method of moments) and using Lemma 5 and then by taking expectations and limits, we get

$$2\,E(\beta_1) - 2\frac{p}{k} \cdot \sum_{i=1}^{k} E(y_i) = p + E((x_1+\ldots+x_k)^2) - 2\frac{p}{k} \sum_{i=1}^{k} (1-\varphi_i(0)) \qquad \text{(EQ 7)}$$

(Note that one must use $\Delta^2(\beta_1^n) = \Delta(\beta_1^n)$ and $\beta_1^n \cdot \Delta(\beta_1^n) = \beta_1^n$, also Lemma 1, to get (EQ 7)).

Since $x_i^2=x_i$ $\forall$ i and x_i, x_j are independent for $i \neq j$ from Lemma 1, we get

$$E((x_1+\ldots+x_k)^2)=k\ E(x_1)+2\frac{k(k-1)}{2}\ E^2(x_1)$$

$$=p+k(k-1)\ \frac{p^2}{k^2} \qquad \text{(EQ 8)}$$

By symmetry, $E(y_i)$ is the same $\forall$ i and this also holds for $y_i(0)$, hence by (EQ 7):

Lemma 7

$$2\ E(\beta_1)-2pE(y_1)=\frac{k-1}{k}\ p^2+2p\ \varphi_1(0) \qquad \text{(EQ 9)} \quad \Box$$

By noting that the r.v. y_1 is the distribution of the size of queue Q_1 as seen by an arriving packet from input 1, we can get a worst case bound for this distribution (leading to an upper bound for $E(\beta_1)$) by using the "bulk arrival" distribution

Y_1= the size of β_1 given that

$x_1=1$ or $x_2=1$ or ... or $x_k=1$

and $\Phi_1(\alpha)=\text{Prob}\{Y_1=\alpha\}$

Clearly (as we will show immediately)

$$E(Y_1) \geq E(y_1)$$

leading to

$$2\ E(\beta_1)\leq \frac{k-1}{k}\ p^2+2p\ \varphi_1(0)+2pE(Y_1) \qquad \text{(EQ 10)}$$

We are going to use an Operational Approach ([BD, 78]) argument to get an upper bound on $E(Y_1)$. Let us consider the case where t_0 is an instant at which arrivals start to happen at Q_1 (no such arrivals happened at t_0-1). Arrivals will continue to happen for, say, λ cycles and then no arrivals will happen at Q_1 for, say, μ cycles. If we assume that the mean number of arrivals per cycle is 1 for the interval of λ cycles, then we get a worst case bound for Y_1. (See Figure 3). Let q_0 be the queue size of Q_1 at t_0.

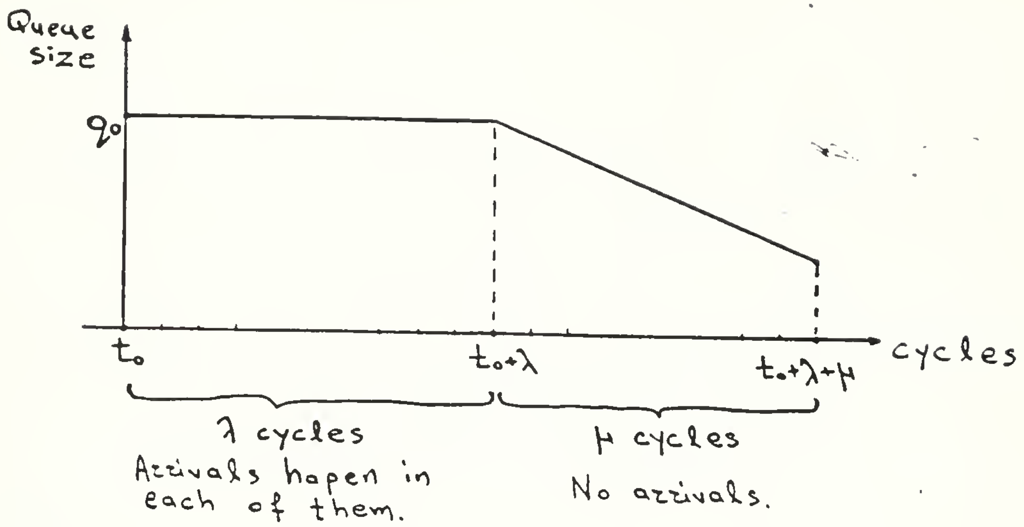

Figure 3: Operational Approach argument.

Clearly $E(Y_1)=q_0-\frac{\mu}{\lambda+1}$ in the interval $(t_0,t_0+\lambda+\mu]$

while

$$E(\beta_1)=\frac{\lambda q_0+\mu q_0-(1+2+\ldots+\mu)}{\lambda+\mu}=q_0-\frac{\mu(\mu+1)}{2(\lambda+\mu)}$$

in the same interval.

$$\text{So} \quad E(Y_1)=E(\beta_1)+\left(\frac{\mu}{\lambda+\mu}\right)\cdot\left(\frac{\lambda-1}{\lambda+1}\right)\cdot\left(\frac{\mu-1}{2}\right) \qquad \text{(EQ 11)}$$

But $\frac{\mu}{\lambda+\mu}\leq 1$, $\frac{\lambda-1}{\lambda+1}\leq 1$, hence

$$E(Y_1)\leq E(\beta_1)+E\left(\frac{\mu-1}{2}\right) \quad \text{i.e.}$$

$$E(Y_1)\leq E(\beta_1)+\frac{E(\mu)-1}{2} \qquad \text{(EQ 12)}$$

Since the $x_1,\ldots,x_k$ are mutually independent and since Prob $\{x_i=0\}=1-p/k$ we get

$$\text{Prob}\{\text{no arrivals at a cycle}\}=\left(1-\frac{p}{k}\right)^k$$

and, hence,

$$E(\mu)=\left(1-\frac{p}{k}\right)^{-k} \qquad \text{(EQ 13)}$$

We conclude

Lemma 8

$$E(Y_1)\leq E(\beta_1)+\frac{\left(1-\frac{p}{k}\right)^{-k}-1}{2}$$

By using EQ 10 then, we get

$$E(\beta_1) \leq \frac{\frac{k-1}{k}p^2+p(1-\frac{p}{k})^{-k}-p+2p\,\varphi_1(0)}{2(1-p)}$$

But $\varphi_1(0) \leq$ 1-utilization fo queue Q_1, since

$$E(Y_1) \geq E(\beta_1)$$

So

$$E(\beta_1) \leq \frac{\frac{k-1}{k}p^2+p(1-\frac{p}{k})^{-k}-p+2p(1-p)}{2(1-p)} \quad \text{i.e.}$$

Lemma 9

$$E(\beta_1) \leq p+\frac{p^2(1-\frac{1}{k})}{2(1-p)}+\frac{p[(1-\frac{p}{k})^{-k}-1]}{2(1-p)} \qquad \text{(EQ 14)}$$

i.e. the mean queue size at the second stage is upper bounded by the sum of the mean queue size of the first stage and the factor

$$\frac{p[(1-\frac{p}{k})^{-k}-1]}{2(1-p)}$$

☐

The result of Lemma 9 has been validated by our simulations. For small p it gives an excellent agreement.

5. The queueing delays at subsequent stages

Let us now inductively assume that, for each output queue Q of stage m, we have for the steady state queue size q_m that $E(\Delta(q_m))=p$ and that (EQ 14) holds for that queue. This holds for stage 2, as we showed in Chapter 4. This implies that $E(x_i)=\frac{p}{k}$ for the corresponding x_i feeding the next stage. Since Lemma 1 holds for all stages and since the rest of the analysis of Chapter 4 (especially EQ 10,11,12,13) do not assume anything else about the $x_1,\ldots,x_k$ except that of Lemma 1, we conclude that we can validate the rest of the equations completely and thus prove for any queue Q' of stage m+1 that its steady state q_{m+1} satisfies $E(\Delta(q_{m+1}))=p$ and also (EQ 14). By induction then, we get:

Lemma 10 (For traffic from processors to memories)

(a) The mean size, $E(\beta)$, of each output queue of any switch of any stage $i \geq 2$, satisfies

$$E(\beta) \leq p + \frac{p^2(1-\frac{1}{k})}{2(1-p)} + \frac{p[(1-\frac{p}{k})^{-k}-1]}{2(1-p)}$$

(b) The mean queueing delay of a packet in any stage $i \geq 2$ satisfies

$$W_{queue} \leq \frac{p(1-\frac{1}{k})}{2(1-p)} + \frac{(1-\frac{p}{k})^{-k}-1}{2(1-p)}$$

Remarks

Our simulations validate Lemma 10. Indeed mean queue sizes increase from the first to the second stage and change insignificantly thereafter. See tables of Appendix. The extention of our results to the case of queues of finite size is currently under investigation. Although researches have argued that finite buffers behave as in the case of infinite buffers when their size exceeds the mean queue length, (see [KS, 83]), it has been remarked (in the design of RP3) that this is not the case for large p. We pose this as an open problem.

REFERENCES

[B, 76] Buzen J.P., "Fundamental Operational Laws of Computer Systems perfomance", Acta Informatica 7, 167-182, 1976.

[BD, 78] Buzen J.P. and Denning P.J., "The Operational Analysis of Computer Network Models", Computing Surveys Vol. 10, No 3, Sept. 1978.

[DJ, 81] Dias D.M. and Jump J.R., "Packet switching interconnection networks for modular systems", Computer 14, Dec. 1981.

[G, 84] Gottlieb A. Private Communication

[GGKMRS,82] Gottlieb A., Grishman R., Kruskal C., McAuliffe K., Rudolph L., Snir M., "The NYU Ultracomputer-Designing a MIMD, Shared-Memory Parallel Machine", 10th Ann. Symp. on Computer Architecture, 1982.

[GL, 73] Goke G. and Lipovski G., "Banyan networks for partitioning multiprocessor systems", 1st Ann Symp. on Computer Architecture, 21-28, 1973.

[KLEI, 75] Kleinrock L., "Queueing Systems", Vol. 1, Wiley 1975.

[KS, 83] Kruskal C.P. and Snir M., "The Performance of Multistage Interconnection Networks for Multiprocessors", IEEE Trans. on Comp., Vol. C-32, No 12, Dec. 1983.

[KSW, 84] Kruskal C.P., Snir M. and Weiss A., Unpublished T.R., U. of Urbana-Champaign, ILL.

[P, 81] Patel J.A., "Performance of processor-memory interconnections for multiprocessors", IEEE Trans. Comp. C-30, 1981.

APPENDIX

Our self-driven simulator is written in Pascal and runs on a VAX/750. We use a three-dimensional array to represent the network. The first dimension is the stage, the second is the switch in the stage and the third is the input of the switch. For the interconnection between stages we use the formula derived in [P, 81]. Our random number generator is of the modulo type.

TABLES

Table 1

Number of processors = 512 (9 stages)

2x2 switches
probability $p=0.2$

steps of the simulation = 20,000

stage #	mean queue length, $E(\beta_1)$	conditional mean queue length, $E(y_1)$
1	0.21370	0.281040
2	0.21185	0.285426
3	0.21445	0.305997
4	0.21035	0.304020
5	0.21215	0.284211
6	0.21850	0.303797
7	0.22010	0.317907
8	0.21300	0.295908
9	0.21390	0.305008

Table 2

Number of processors = 512 (9 stages)
2x2 switches
probability p=0.4
steps of the simulation = 20,000

stage #	mean queue length, $E(\beta_1)$	conditional mean queue length, $E(y_1)$
1	0.46525	0.549005
2	0.47075	0.596522
3	0.47620	0.615155
4	0.48165	0.638999
5	0.47375	0.615714
6	0.47385	0.620073
7	0.49700	0.646475
8	0.48940	0.652619
9	0.48715	0.627983

Table 3

Number of processors = 512 (9 stages)
2x2 switches
probability p=0.6
steps of the simulation = 20,000

stage #	mean queue length, $E(\beta_1)$	conditional mean queue length, $E(y_1)$
1	0.82845	0.959728
2	0.87660	1.074090
3	0.83835	1.031580
4	0.84580	1.056960
5	0.86805	1.081350
6	0.86185	1.084420
7	0.82795	1.015700
8	0.86145	1.053790
9	0.85410	1.046780

NYU COMPSCI TR-289 c.1
Bouras, C
Queueing delays in buffered multistage interconnection networks.

NYU COMPSCI TR-289 c.1
Bouras, C
Queueing delays in buffered multistage interconnection networks.

This book may be kept
FOURTEEN DAYS
A fine will be charged for each day the book is kept overtime.

NOV 3 0 1987

GAYLORD 142 PRINTED IN U.S.A.